

# TASK 6: Sales Trend Analysis Using Aggregations

## 1. Use EXTRACT(MONTH FROM order\_date) for month.

SELECT

```
EXTRACT(YEAR FROM order_date) AS order_year,  
EXTRACT(MONTH FROM order_date) AS order_month,  
SUM(total_revenue) AS total_monthly_revenue
```

FROM online\_sales

GROUP BY order\_year, order\_month;

The screenshot shows a database tool interface. The 'Result Grid' tab is active, displaying the results of the first query. The data is as follows:

order_year	order_month	total_monthly_revenue
2024	1	14548.32
2024	2	10803.37
2024	3	12849.24
2024	4	12451.69
2024	5	8455.49
2024	6	7384.55
2024	7	6797.08
2024	8	7278.11

The 'Action Output' tab is also visible, showing the execution of the queries. The third action (SELECT) returned 9 rows.

## 2. GROUP BY year/month.

SELECT

```
EXTRACT(YEAR FROM order_date) AS order_year,  
EXTRACT(MONTH FROM order_date) AS order_month,  
COUNT(DISTINCT transaction_id) AS order_volume
```

FROM online\_sales

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

The screenshot shows the same database tool interface. The 'Result Grid' tab is active, displaying the results of the second query. The data is as follows:

order_year	order_month	order_volume
2024	1	31
2024	2	29
2024	3	31
2024	4	30
2024	5	31
2024	6	30
2024	7	31
2024	8	27

The 'Action Output' tab shows the execution of the queries. The third action (SELECT) returned 9 rows, and the fourth action (SELECT) also returned 9 rows.

### 3. Use SUM() for revenue.

SELECT

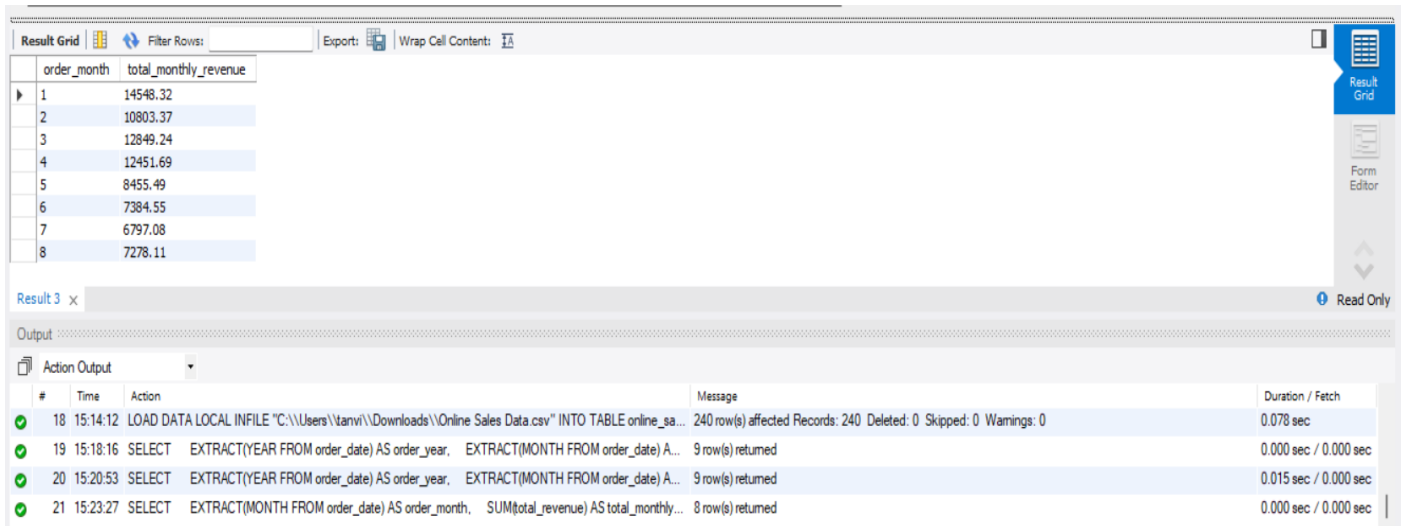
EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(total\_revenue) AS total\_monthly\_revenue

FROM online\_sales

GROUP BY order\_month

ORDER BY order\_month;



order_month	total_monthly_revenue
1	14548.32
2	10803.37
3	12849.24
4	12451.69
5	8455.49
6	7384.55
7	6797.08
8	7278.11

#	Time	Action	Message	Duration / Fetch
18	15:14:12	LOAD DATA LOCAL INFILE "C:\\Users\\tanvi\\Downloads\\Online Sales Data.csv" INTO TABLE online_sa...	240 row(s) affected Records: 240 Deleted: 0 Skipped: 0 Warnings: 0	0.078 sec
19	15:18:16	SELECT EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A...	9 row(s) returned	0.000 sec / 0.000 sec
20	15:20:53	SELECT EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A...	9 row(s) returned	0.015 sec / 0.000 sec
21	15:23:27	SELECT EXTRACT(MONTH FROM order_date) AS order_month, SUM(total_revenue) AS total_monthly...	8 row(s) returned	0.000 sec / 0.000 sec

### 4. COUNT(DISTINCT order\_id) for volume.

SELECT

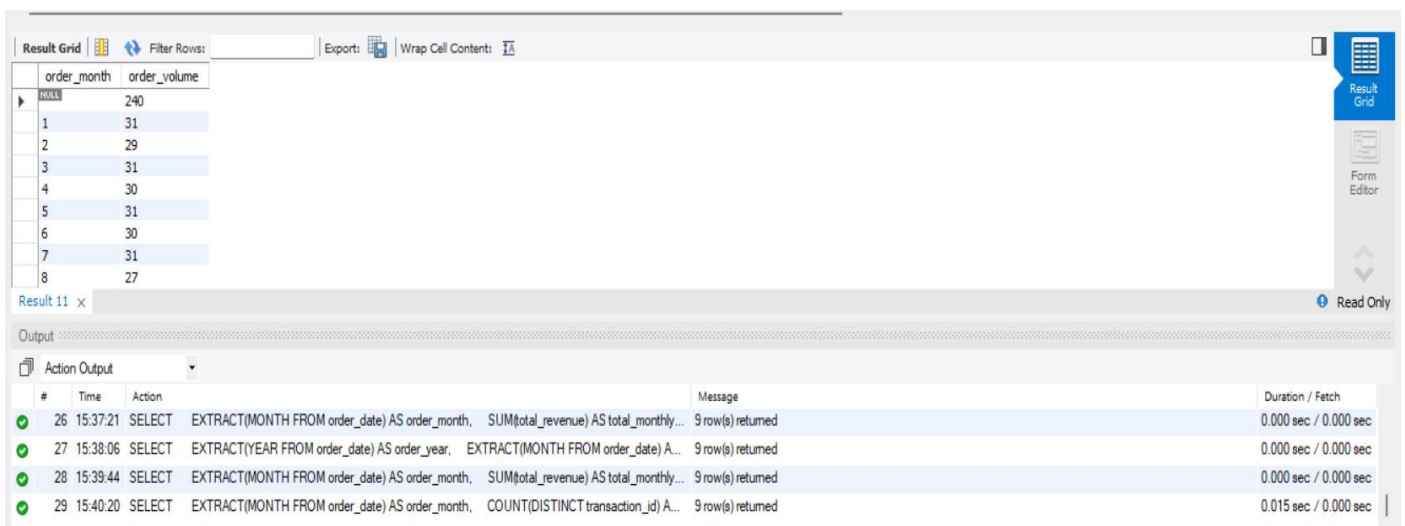
EXTRACT(MONTH FROM order\_date) AS order\_month,

COUNT(DISTINCT transaction\_id) AS order\_volume

FROM online\_sales

GROUP BY order\_month

ORDER BY order\_month;



order_month	order_volume
1	240
2	31
3	29
4	31
5	30
6	31
7	30
8	31

#	Time	Action	Message	Duration / Fetch
26	15:37:21	SELECT EXTRACT(MONTH FROM order_date) AS order_month, SUM(total_revenue) AS total_monthly...	9 row(s) returned	0.000 sec / 0.000 sec
27	15:38:06	SELECT EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A...	9 row(s) returned	0.000 sec / 0.000 sec
28	15:39:44	SELECT EXTRACT(MONTH FROM order_date) AS order_month, SUM(total_revenue) AS total_monthly...	9 row(s) returned	0.000 sec / 0.000 sec
29	15:40:20	SELECT EXTRACT(MONTH FROM order_date) AS order_month, COUNT(DISTINCT transaction_id) A...	9 row(s) returned	0.015 sec / 0.000 sec

5. Use ORDER BY for sorting.

```
SELECT
    EXTRACT(YEAR FROM order_date) AS order_year,
    EXTRACT(MONTH FROM order_date) AS order_month,
    SUM(total_revenue) AS total_monthly_revenue
FROM online_sales
GROUP BY order_year, order_month
ORDER BY total_monthly_revenue DESC;
```

Result Grid

	order_year	order_month	total_monthly_revenue
▶	2024	1	14548.32
	2024	3	12849.24
	2024	4	12451.69
	2024	2	10803.37
	2024	5	8455.49
	2024	6	7384.55
	2024	8	7278.11
	2024	7	6797.08

Result 6 x

Output

#	Time	Action	Message	Duration / Fetch
21	15:23:27	SELECT	EXTRACT(MONTH FROM order_date) AS order_month, SUM(total_revenue) AS total_monthly... 8 row(s) returned	0.000 sec / 0.000 sec
22	15:25:29	SELECT	EXTRACT(MONTH FROM order_date) AS order_month, COUNT(DISTINCT transaction_id) A... 8 row(s) returned	0.016 sec / 0.000 sec
23	15:26:56	SELECT	EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A... 5 row(s) returned	0.000 sec / 0.000 sec
24	15:27:15	SELECT	EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A... 9 row(s) returned	0.000 sec / 0.000 sec

6. Limit results for specific time periods.

```
SELECT
    EXTRACT(YEAR FROM order_date) AS order_year,
    EXTRACT(MONTH FROM order_date) AS order_month,
    SUM(total_revenue) AS total_monthly_revenue,
    COUNT(DISTINCT transaction_id) AS order_volume
FROM online_sales
WHERE order_date BETWEEN '2024-03-01' AND '2024-06-30'
GROUP BY order_year, order_month
ORDER BY order_year, order_month;
```

Result Grid

	order_year	order_month	total_monthly_revenue	order_volume
▶	2024	3	12849.24	31
	2024	4	12451.69	30
	2024	5	8455.49	31
	2024	6	7384.55	30

Result 7 x

Output

#	Time	Action	Message	Duration / Fetch
22	15:25:29	SELECT	EXTRACT(MONTH FROM order_date) AS order_month, COUNT(DISTINCT transaction_id) A... 8 row(s) returned	0.016 sec / 0.000 sec
23	15:26:56	SELECT	EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A... 5 row(s) returned	0.000 sec / 0.000 sec
24	15:27:15	SELECT	EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A... 9 row(s) returned	0.000 sec / 0.000 sec
25	15:30:03	SELECT	EXTRACT(YEAR FROM order_date) AS order_year, EXTRACT(MONTH FROM order_date) A... 4 row(s) returned	0.000 sec / 0.000 sec